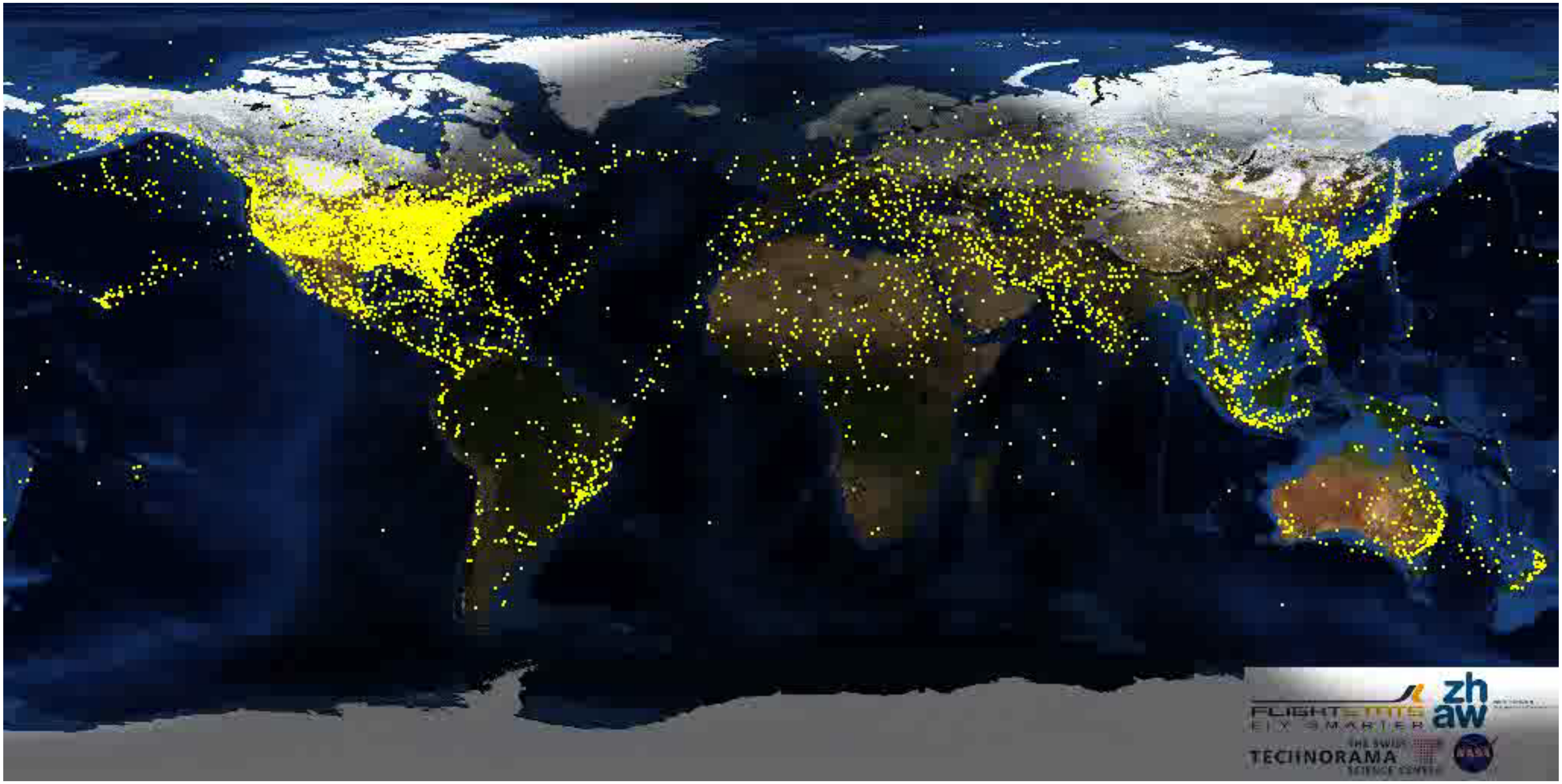
Towards Verification and Validation for Increased Autonomy

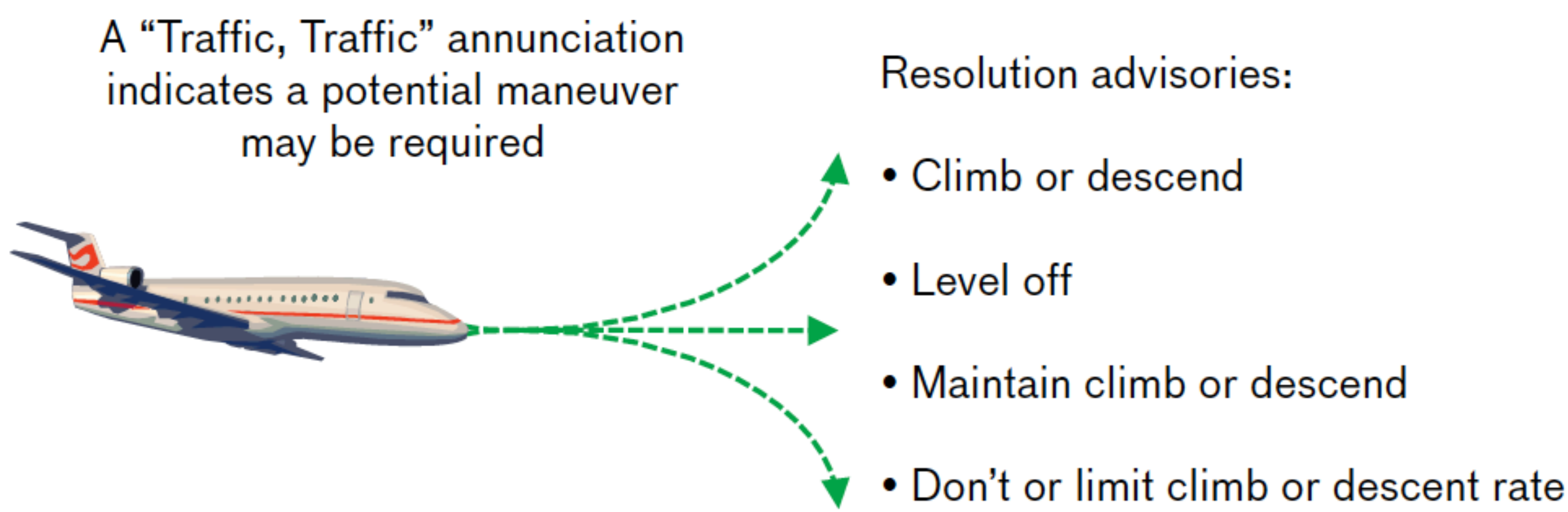
Dimitra Giannakopoulou

an aircraft is not alone in the sky...



Safe Air Transportation

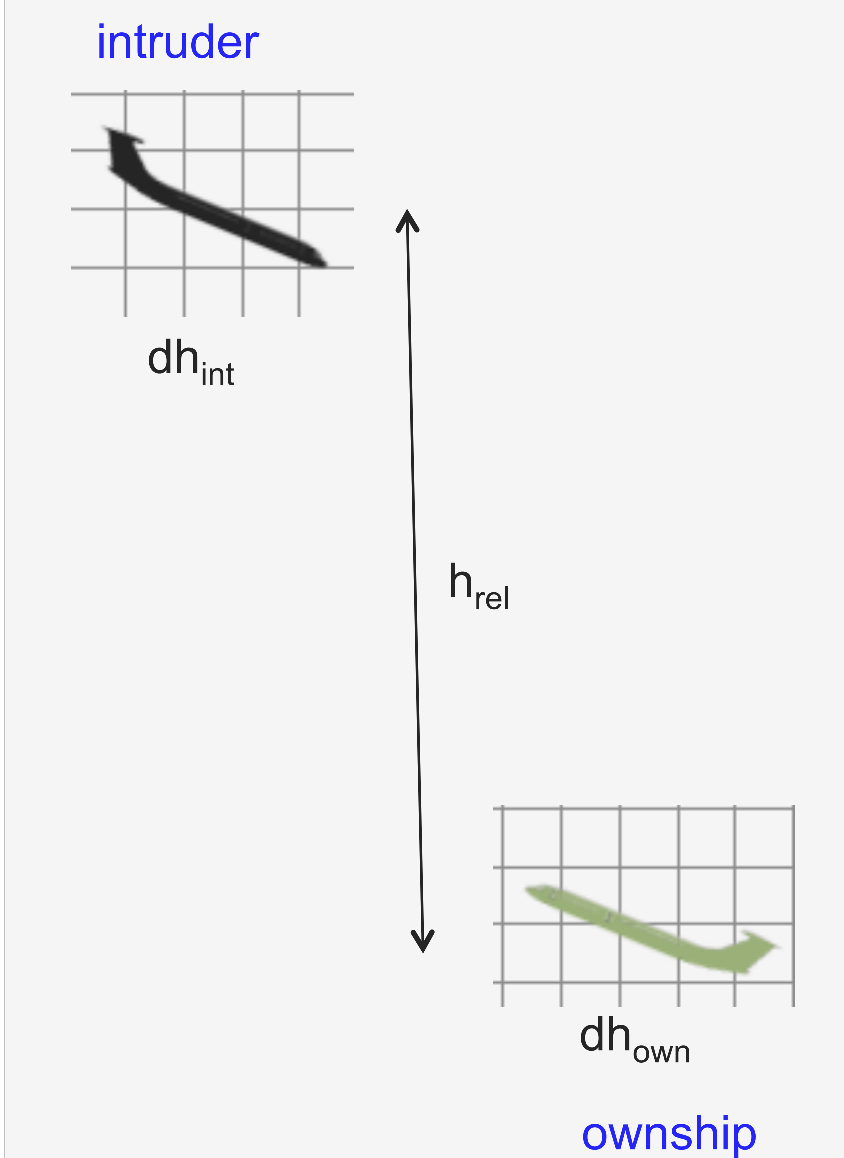
- Air traffic operations are expected to increase significantly. Automation must maintain or exceed current safety standards
- Separation Assurance – algorithms and systems gradually taking the role of air-traffic controllers to enable reduced aircraft separation
- Onboard-Collision Avoidance Systems – TCAS, ACAS X



ACAS X – a completely new paradigm

- 40 secs from Near-Mid-Air Collision (NMAC)
- state variables
  - **h_{rel}** : relative altitude, in [-1000…1000] *ft*
  - **dh_{own}** / **dh_{int}** : ownship / intruder climb rates, in [-2500…2500]*ft/min*
- **a_{prev}** / **s_{RA}**: advisory issued by ACAS X in previous sec / current pilot state, both in {COC, CL/DES1500, SCL/SDES1500, SCL/SDES2500
- update and advisory **frequency** is set to 1 sec
- **discretization resolution** *n* for a variable V means that V is discretized to *n* points above and *n* points below 0 within its interval of values. For example, discretization resolution of 10 for h_{rel} means:

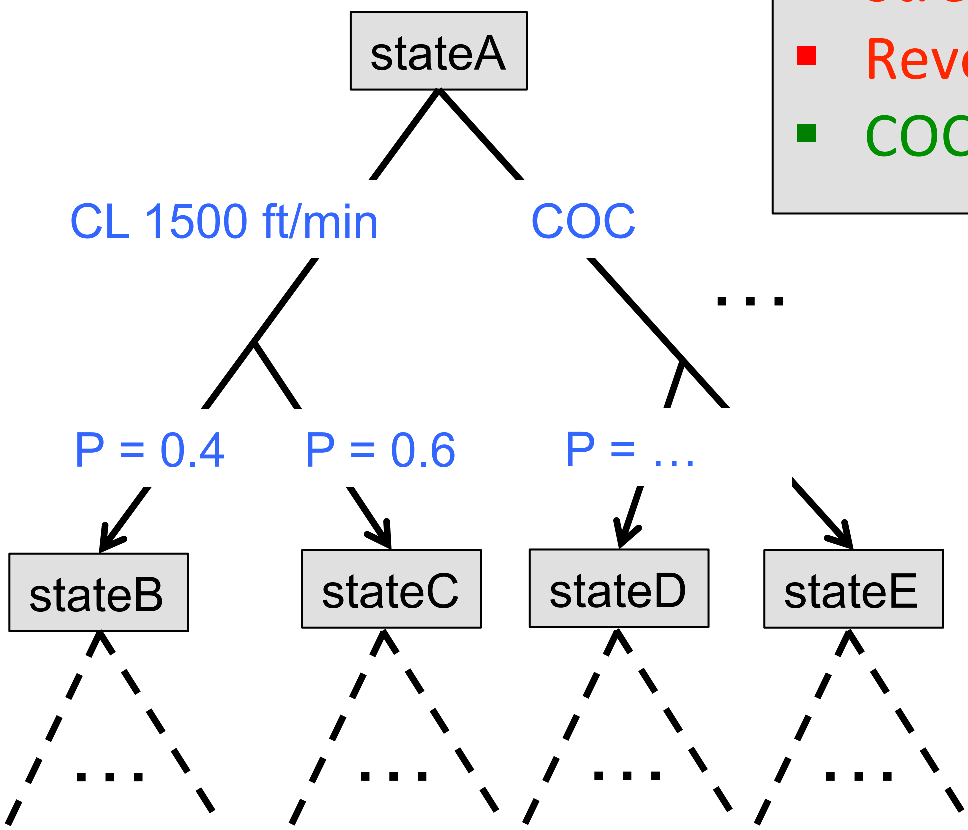
  {-1000, -900, -800, … , 0, 100, …, 900, 1000}



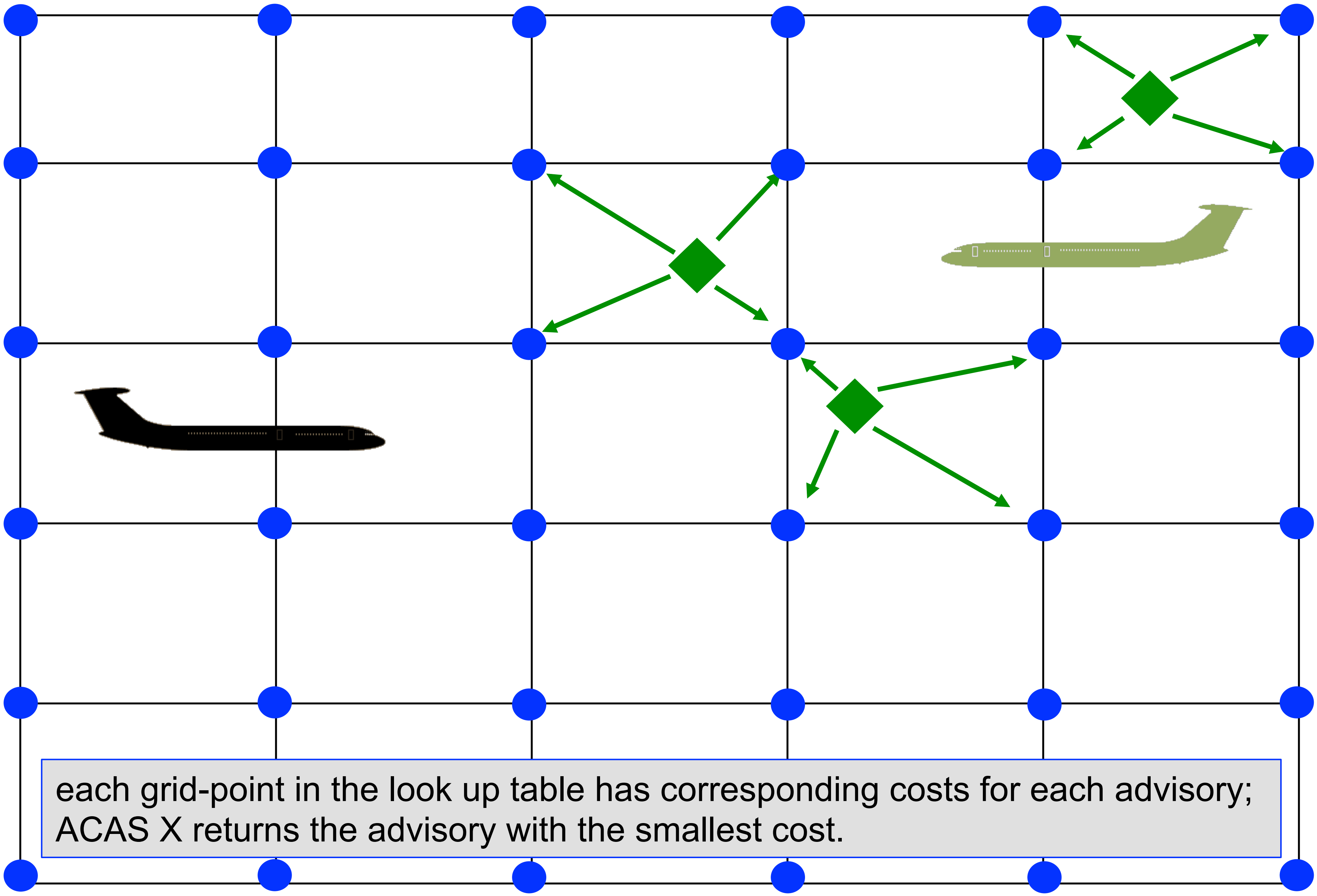
ACAS X – goals

minimize costs / maximize rewards

- NMAC (near-mid-air collision
- Alert (from COC to advisory)
- Strengthening (strengthen advisory)
- Reversal (e.g. climb to descend)
- COC (clear of conflict)

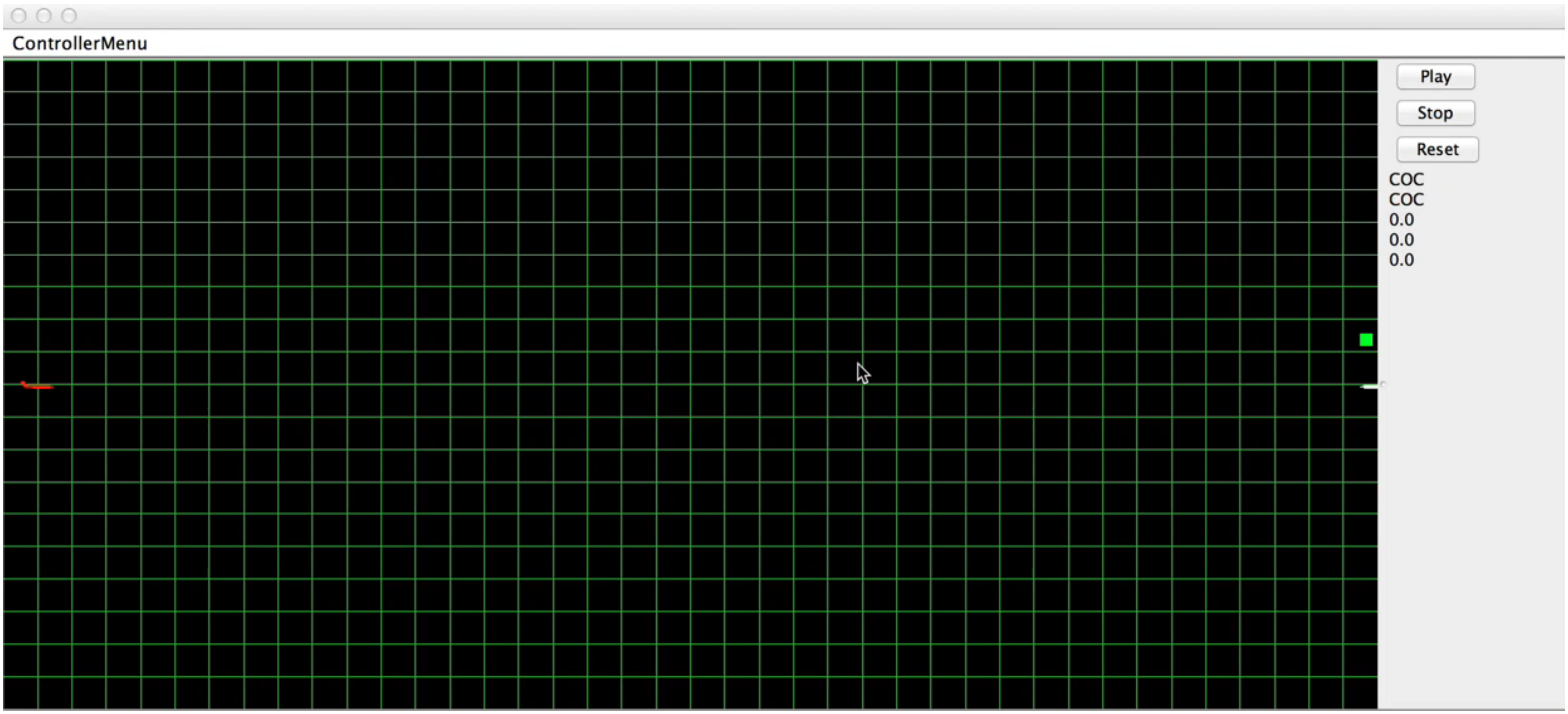


deploying ACAS X as a look up table

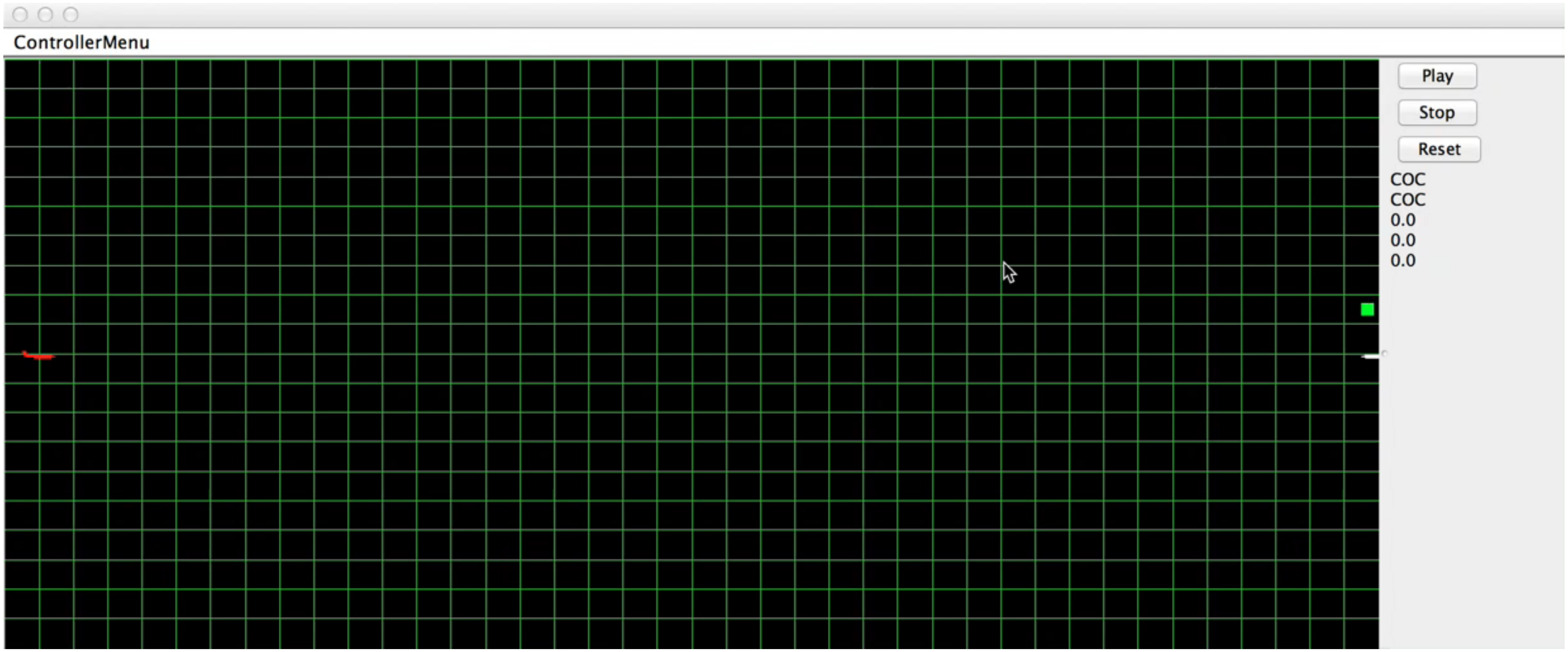


each grid-point in the look up table has corresponding costs for each advisory; ACAS X returns the advisory with the smallest cost.

simulation with low NMAC weight (0.01)

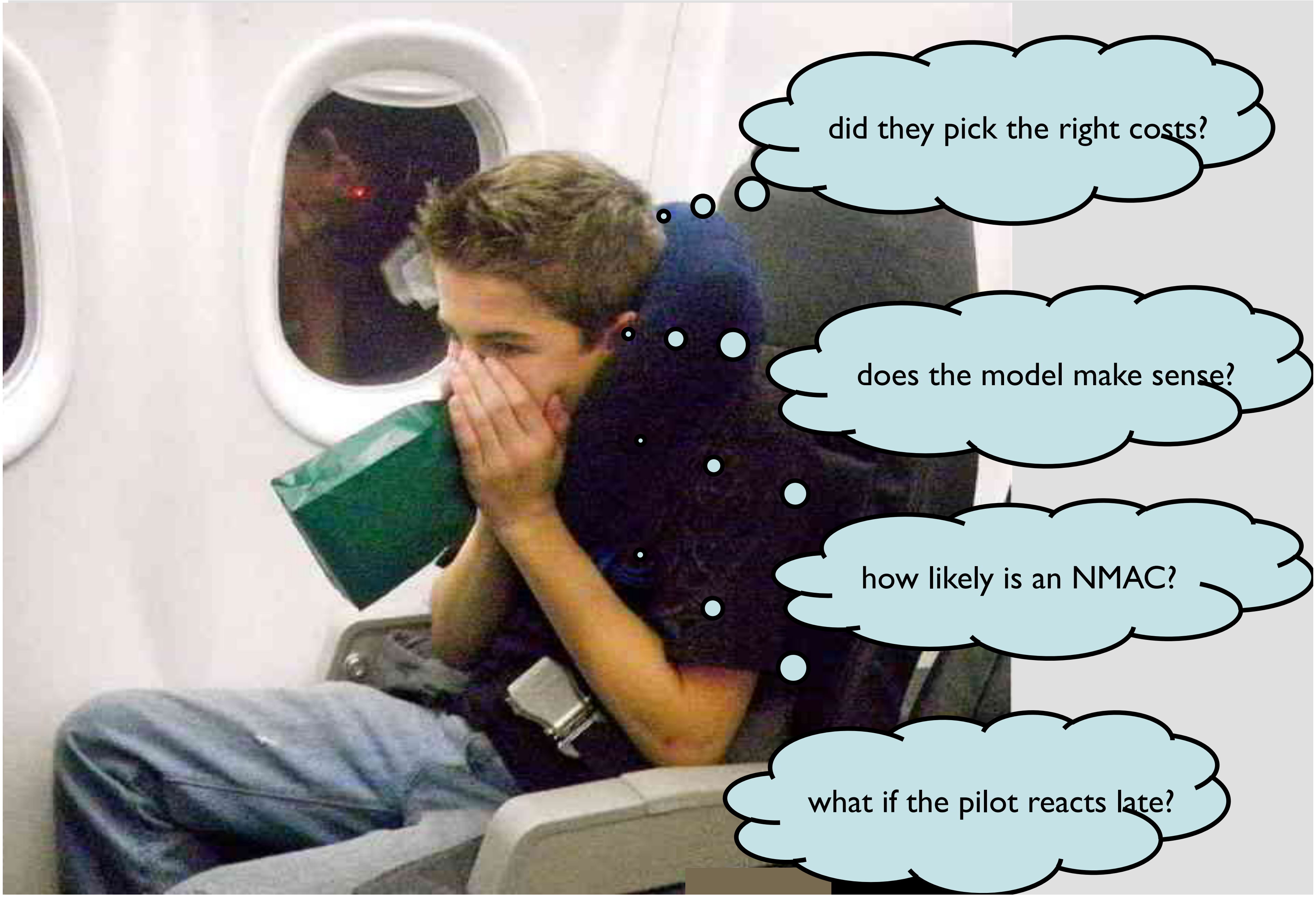


simulation with high NMAC weight (100)



verification starts with asking questions

would you trust ACAS X in a flight?

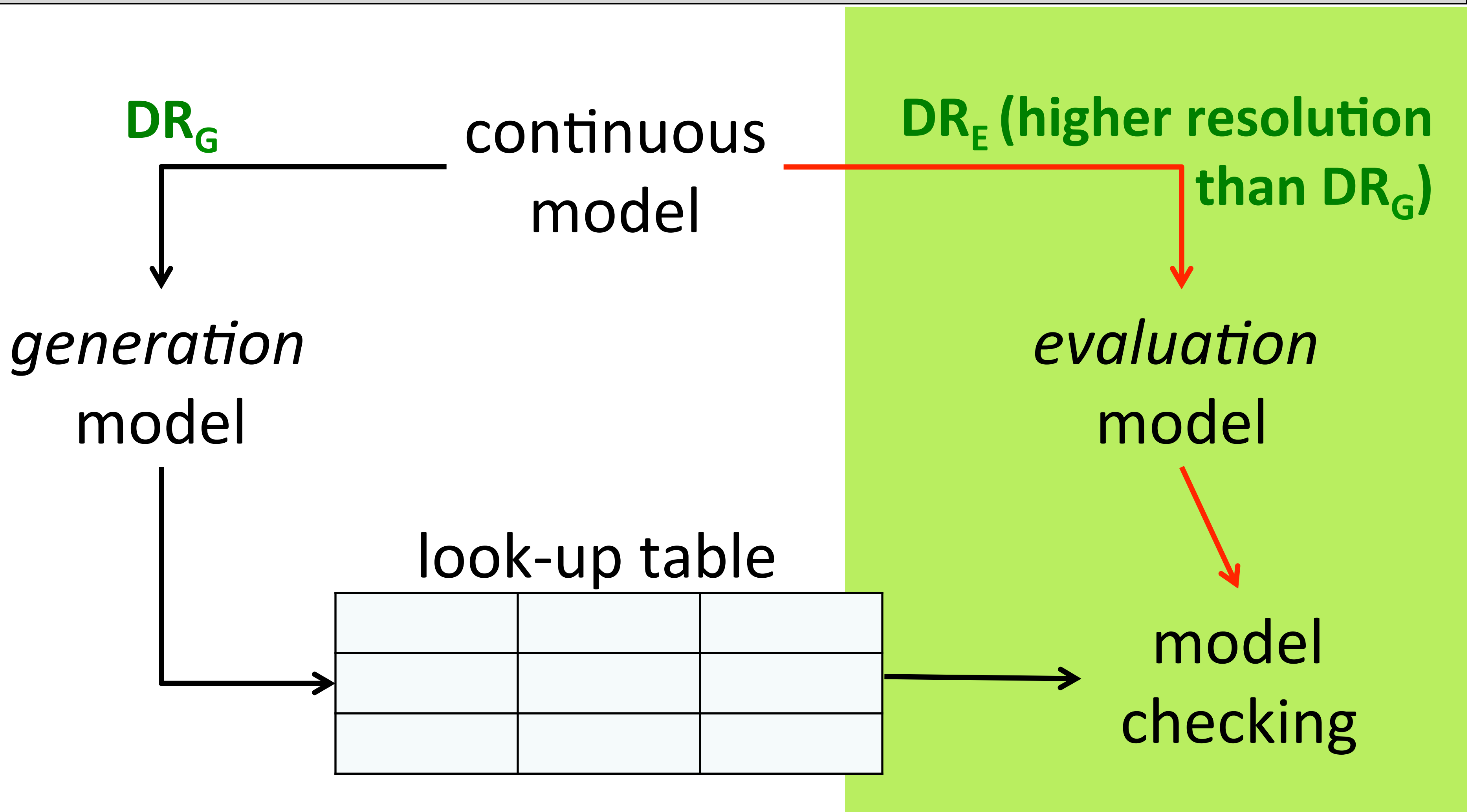


...and proceeds with answering them

discretization resolution (dh_{own}, dh_{int}, h_{rel})

DR_G : model discretization resolution for look-up table generation; baseline [KC 2011] resolution is (dh_{own}=10, dh_{int}=10, h_{rel}=10)

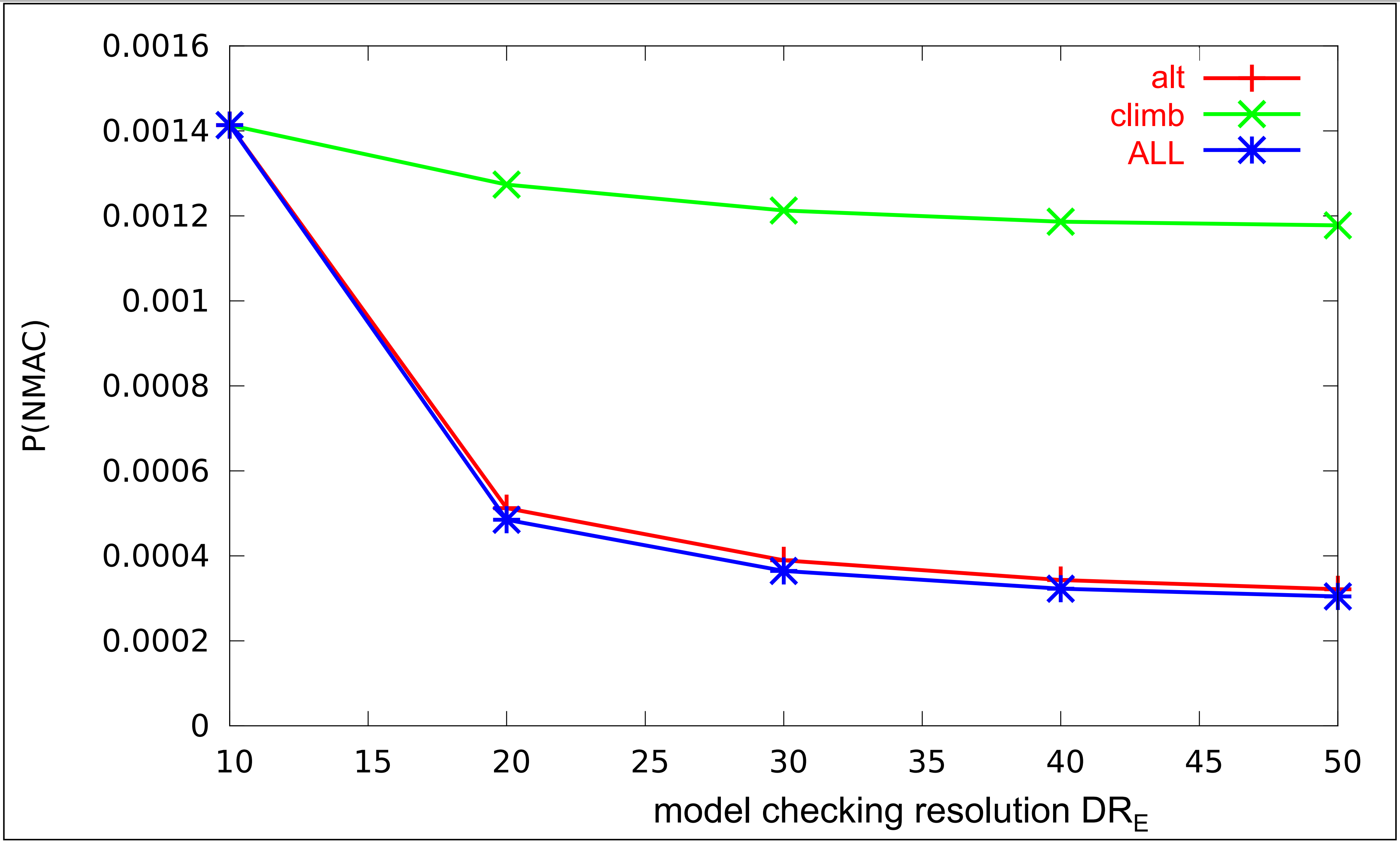
DR_E : model discretization resolution to model check look-up table



effects of resolution DR_E on model checking

- we compute P(NMAC) of the baseline look up table deployed in models that are obtained through discretization with varying resolutions DR_E (dh_{own}, dh_{int}, h_{rel})
  - ALL varies climb rates and relative altitude in DR_E: (20, 20, 20), (30, 30, 30), …
  - climb varies climb rates only in DR_E: (20, 20, 10), (30, 30, 10), …
  - alt varies relative altitude only in DR_E: (10, 10, 20), (10, 10, 30), …

effects of resolution DR_E on model checking



- P(NMAC) decreases with higher evaluation resolutions
- relative altitude discretization is indicative

probabilistic model checking $DR_E = (50, 50, 100)$

allows precise *automated analysis* of probabilistic properties expressed in a formal logic such as *PCTL*; generates *encounters* that exhibit property-related behaviors

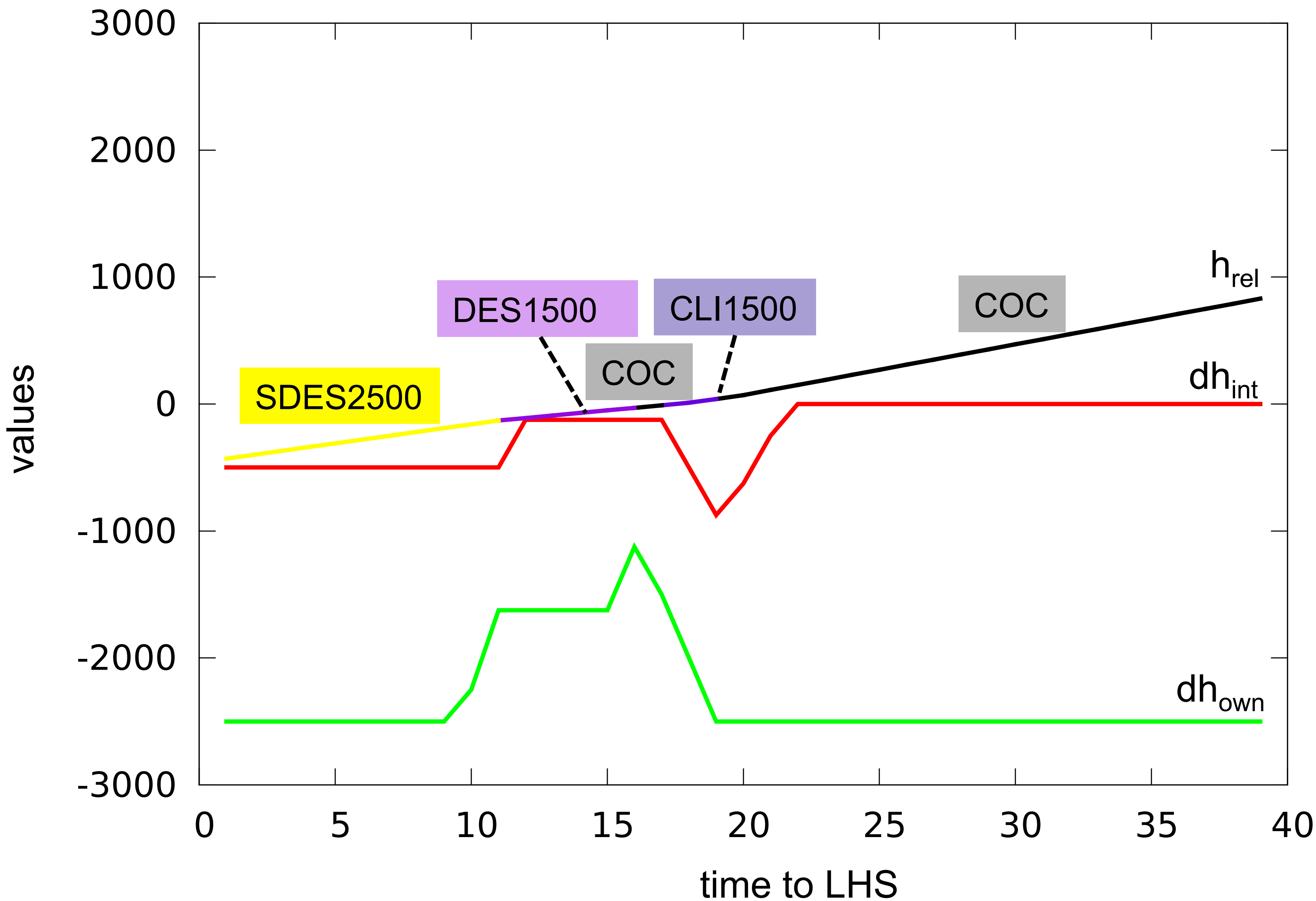
- what is the probability of NMAC? ($\mathbf{P}{=}?\mathbf{[}\mathrm{F\ NMAC}\mathbf{]}$) 2.5 x 10^{-4}
- what if pilot responds immediately?

  ($\mathbf{P}{=}?\mathbf{(}\mathrm{F\ NMAC} \mid \mathrm{G}\mathbf{a_{prev} = s_{RA}}\mathbf{)}$) 2.3 x 10^{-8}

- what is the probability of a split advisory? 1.8 x 10^{-3}

  $\mathbf{P}{=}?\mathbf{[\ F\ (}\,!\mathrm{COC} \wedge \ \mathbf{P}{=}1\mathbf{[X}\ \mathrm{COC}\mathbf{]} \wedge \mathbf{P{>}}0\ \mathbf{[F}\ !\mathrm{COC}\mathbf{]\ )]}$
- split advisories are harder to directly take into account during look up table generation because they require to record history

split advisory encounter

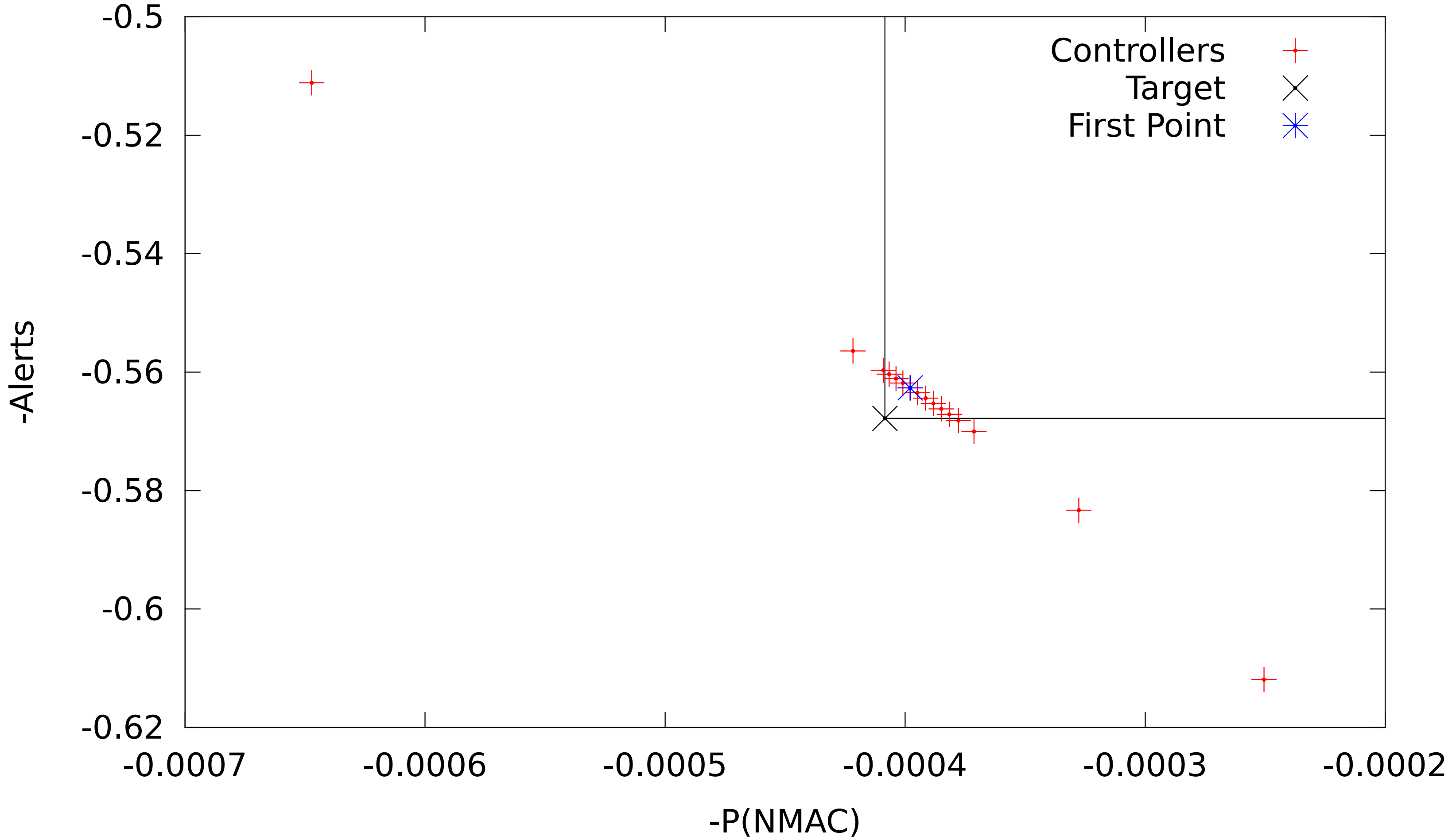


(reward for COC + cost of alert) < cost of reversal ("sneaky" reversals)

synthesis / design

weights vs. performance

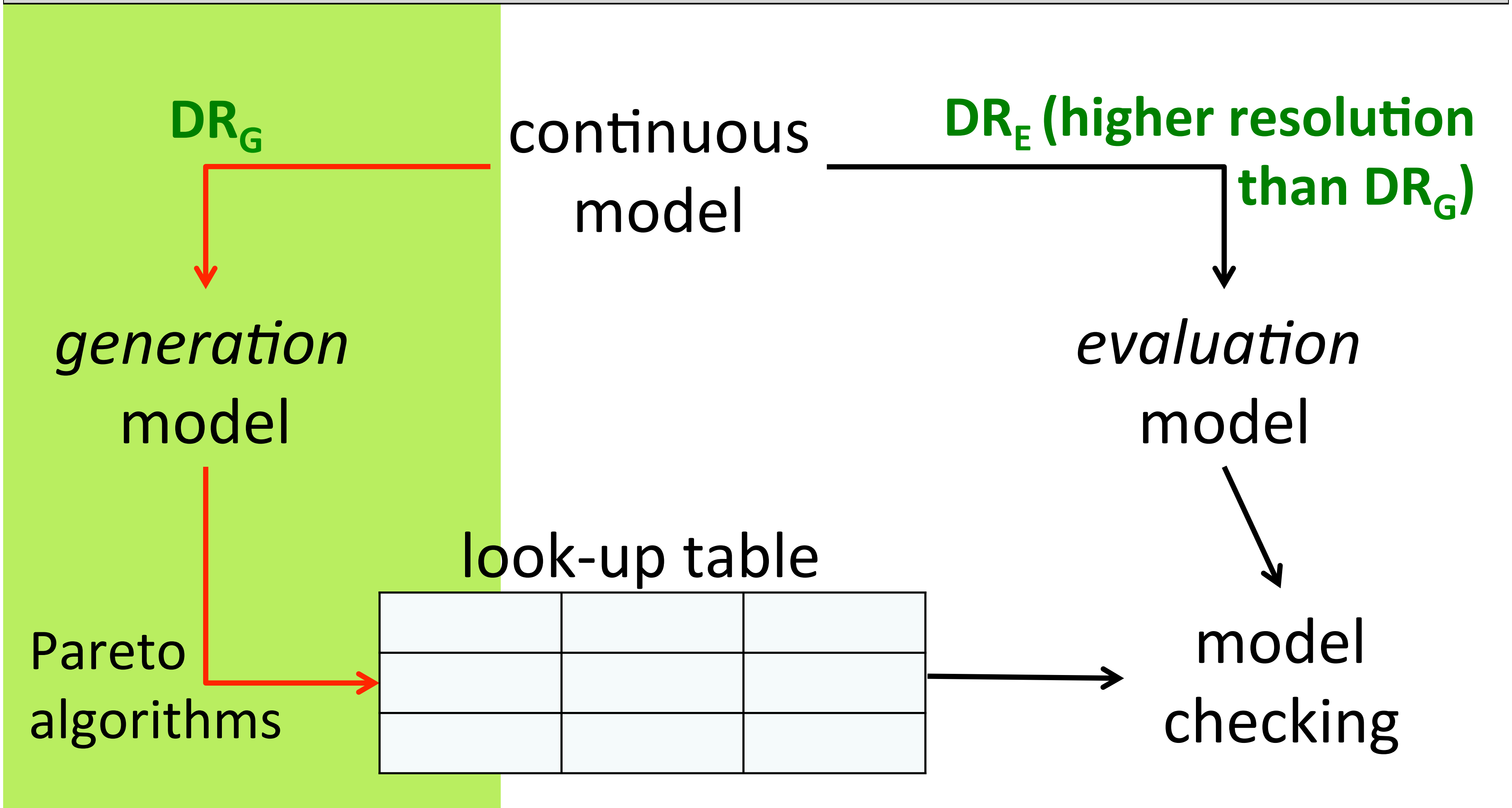
- tune look-up tables based on minimum acceptable performance
  - deterministic look-up tables based on weights form a convex Pareto front; we implement algorithms that approximate it *above* target performance



discretization resolution (dh_{own}, dh_{int}, h_{rel})

DR_G : model discretization resolution for look-up table generation; baseline [KC 2011] resolution is (dh_{own}=10, dh_{int}=10, h_{rel}=10)

DR_E : model discretization resolution to model check look-up table



synthesizing better tables in higher DR_G

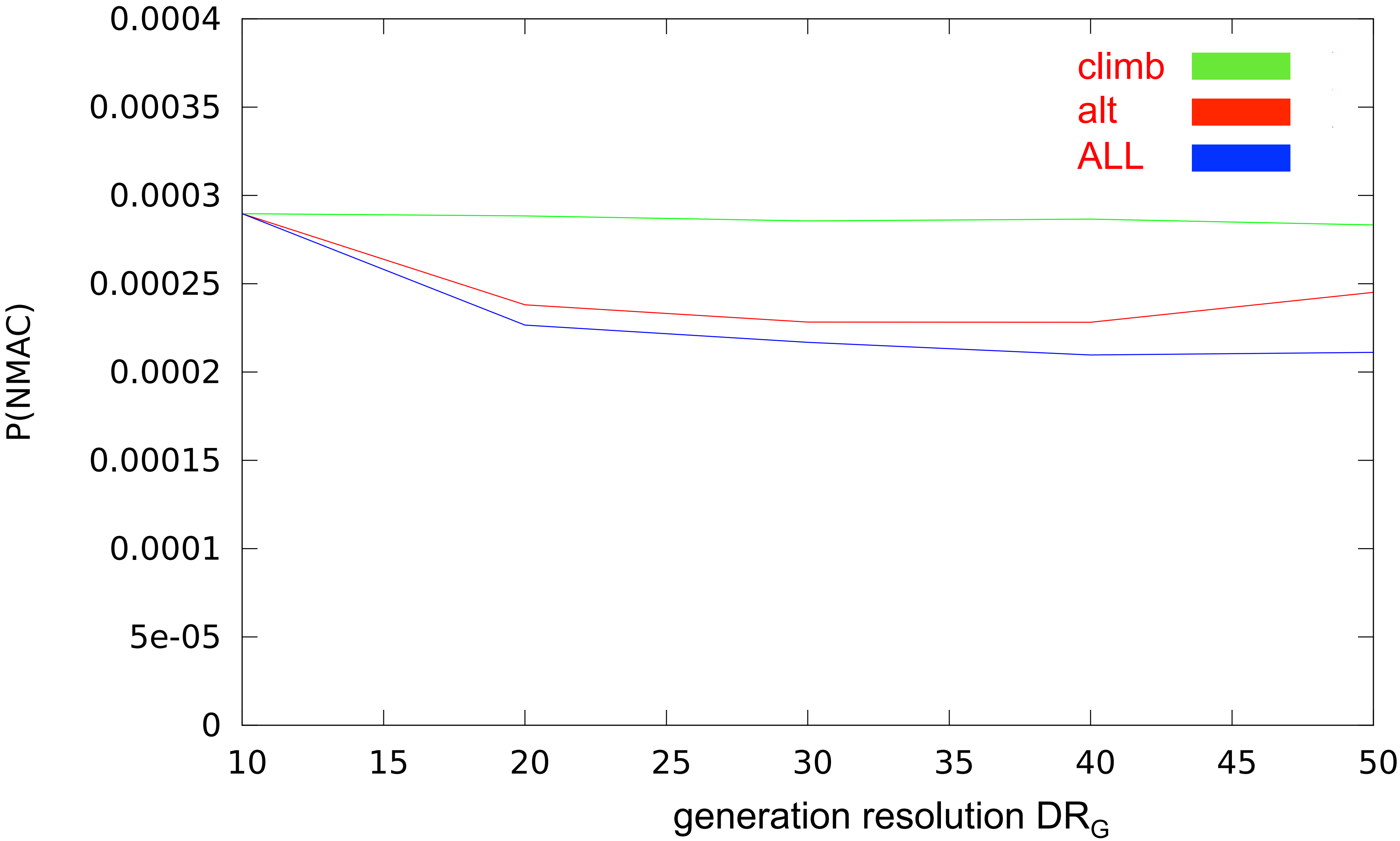
- question: what is the effect of resolution discretization DR_G on look-up table synthesis?

- experiment set up:
  - evaluate baseline (dh_{own}=10, dh_{int}=10, h_{rel}=10) [KC 2011] in new DR_G
  - use result as target for synthesis

- how we vary resolutions DR_G (dh_{own}, dh_{int}, h_{rel})
  - ALL varies climb rates and relative altitude in DR_G: (20, 20, 20), (30, 30, 30), …
  - climb varies climb rates only in DR_G: (20, 20, 10), (30, 30, 10), …
  - alt varies relative altitude only in DR_G: (10, 10, 20), (10, 10, 30), …
  - *note that alt results in the smallest look up tables − in terms of numbers of states − for each value increase*

- compare the synthesized look-up tables in DR_E = (50, 50, 100)

table synthesis at different resolutions



recommendation: (10, 10, 30), or (n, n, 3*n)

verification achievements

- we could not use off-the-shelf tools, so we built VeriCA toolset
  - our tools support models written in Java
  - we customized verification and synthesis algorithms for ACAS X needs
- we analyzed ACAS X version that we reproduced based on:

  Kochenderfer, M. J., and Chryssanthacopoulos, J. P. Robust airborne collision avoidance through dynamic programming. Project Report ATC-371, Massachusetts Institute of Technology, Lincoln Laboratory, 2011.

- ETAPS 2014 EASST best paper award
  - Christian von Essen, Dimitra Giannakopoulou: *Analyzing the Next Generation Airborne Collision Avoidance System,* TACAS 2014.
- FAA / NASA Ames Interagency Agreement for ACAS X and VeriCA
  - thus able to apply our subsequent work on the actual ACAS X code

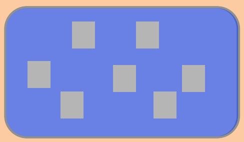
model quality

comparing model to real world

State machine model with probabilistic transitions (MDP) is used to generate onboard look-up table. *The MDP is not present in the onboard system.*

We defined Conformance Relations that compare MDP model to flight data

t



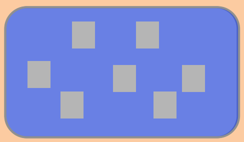
state estimate at time t
on look-up table

comparing model to real world

State machine model with probabilistic transitions (MDP) is used to generate onboard look-up table. *The MDP is not present in the onboard system.*

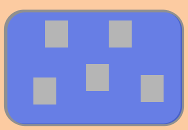
We defined Conformance Relations that compare MDP model to flight data

t



state estimate at time t
on look-up table

t+1sec

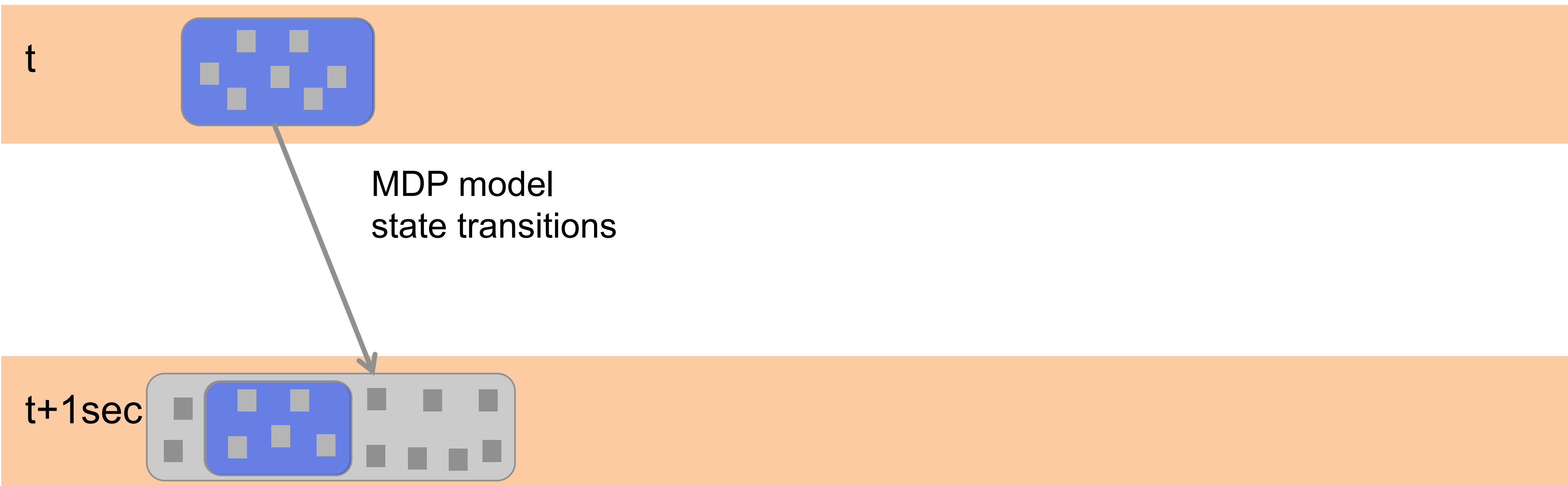


state estimate at time t+1
on look-up table

comparing model to real world

State machine model with probabilistic transitions (MDP) is used to generate onboard look-up table. *The MDP is not present in the onboard system.*

We defined Conformance Relations that compare MDP model to flight data

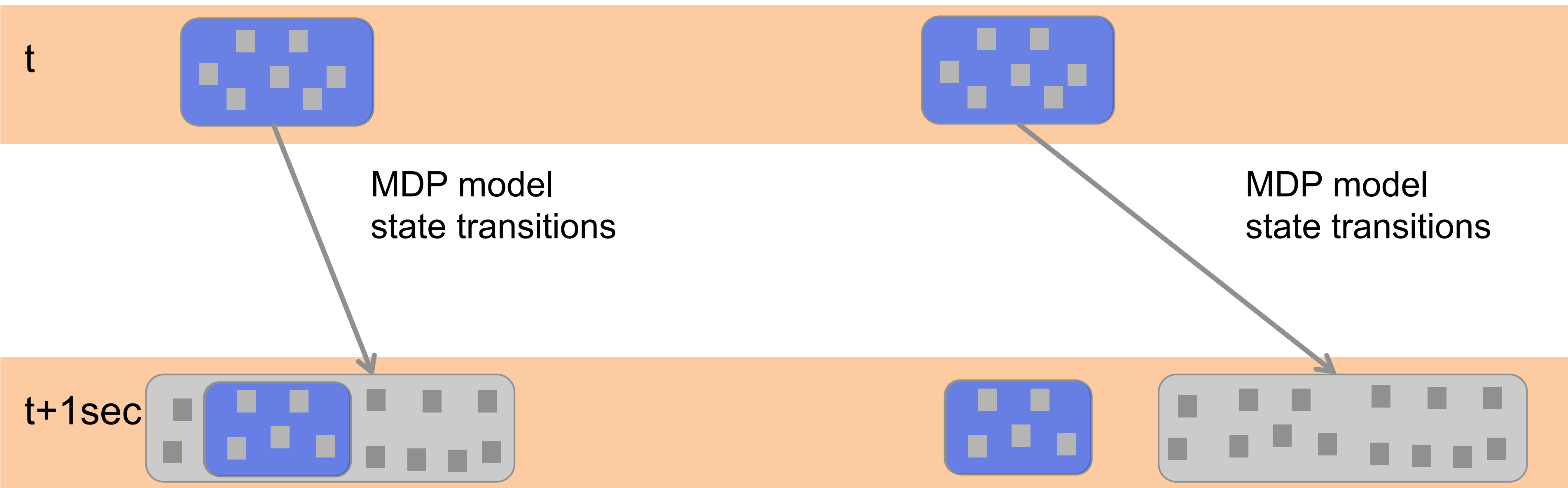


full conformance:
ACAS X states contained
in MDP states

comparing model to real world

State machine model with probabilistic transitions (MDP) is used to generate onboard look-up table. *The MDP is not present in the onboard system.*

We defined Conformance Relations that compare MDP model to flight data



full conformance:
ACAS X states contained
in MDP states

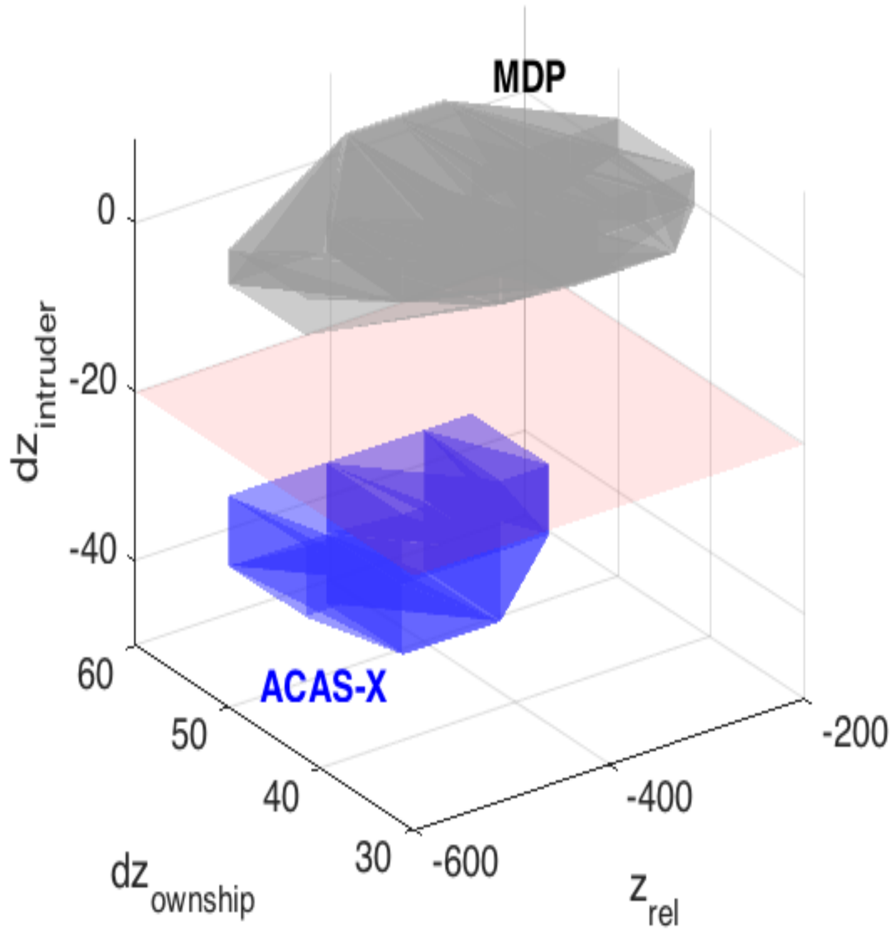
non-conformance:
ACAS X and MDP states
do not intersect

data generation and analysis

Data generation: Non-conforming encounters are rare in test data of the ACAS X distribution. We used a reinforcement learning framework to target generation of non-conforming simulated encounters.

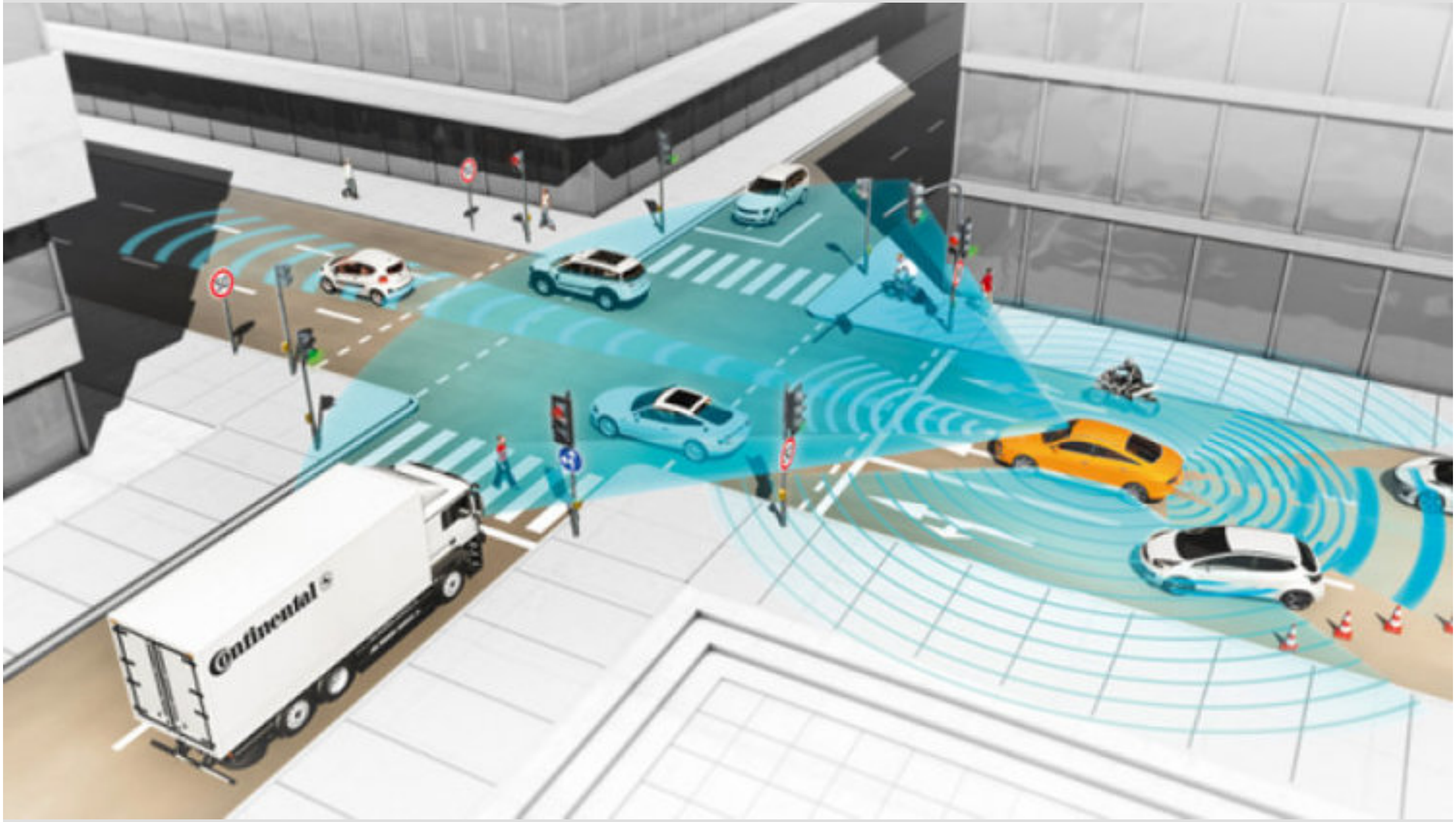
Data Analysis: The intruder climb rate has been identified as a common factor for divergence across the data. *Further analysis is needed.*

Open question: Does non-conformance imply potential violation of safety requirements?



applicability

self-driving cars

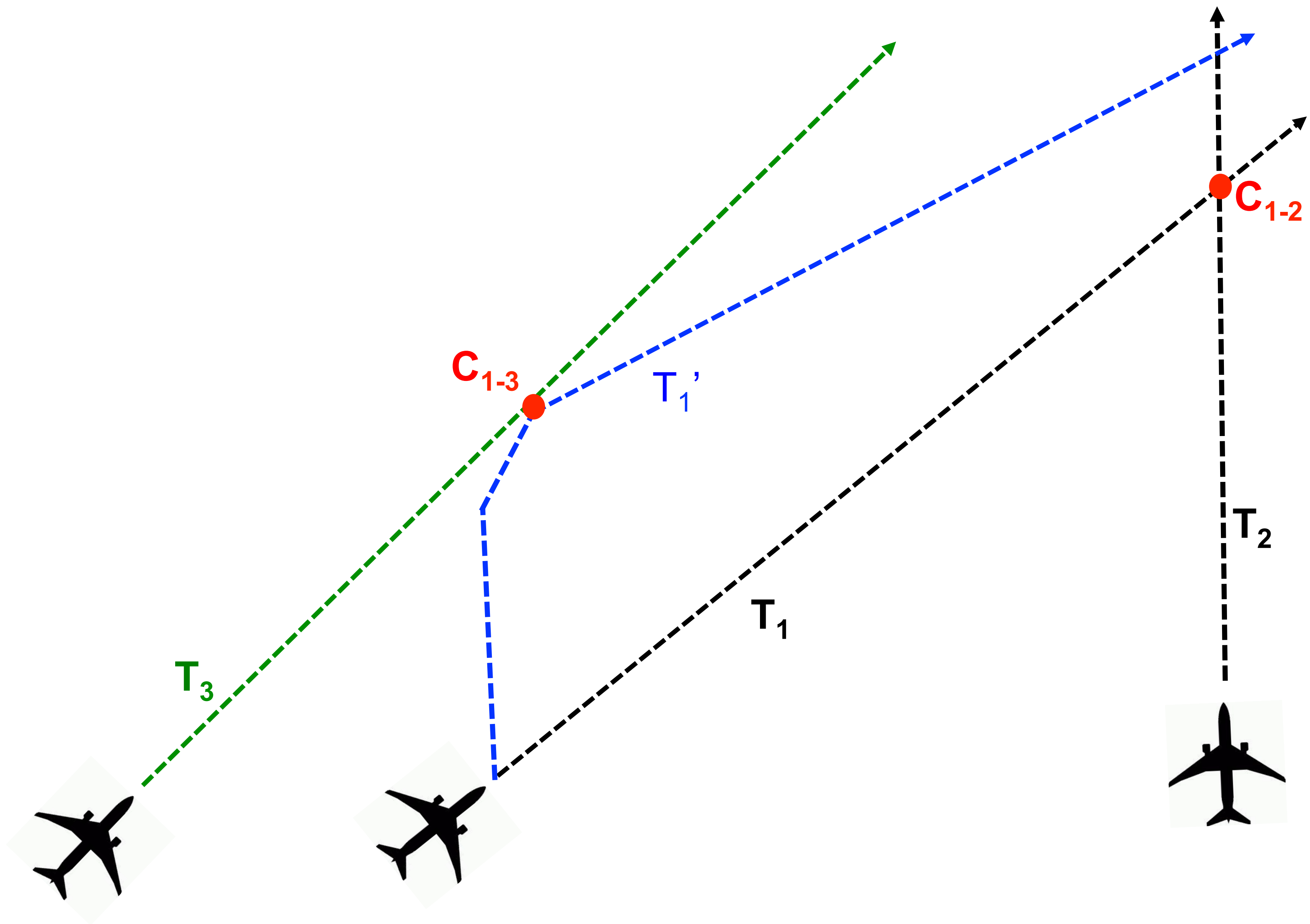


V&V of autonomy

- formulation of requirements is harder – autonomy-specific?
  - optimization, adaptive and learning algorithms
  - example: loss of separation, ACAS X

generate strategic secondary aircraft

no picked resolution is allowed to cause a more imminent secondary conflict



V&V of autonomy

- formulation of requirements is harder – autonomy-specific?
  - optimization, adaptive and learning algorithms
  - example: separation assurance, ACAS X
- need for advanced testing infrastructures
  - test case generation for stress-testing and requirements coverage
  - examples: ACAS X, separation assurance, autonomous vehicles
- V&V at runtime, including requirements
  - ACAS X (error prediction with statistical learning)
  - separation assurance
- trust
  - extensive verification
  - explanation of decision-making algorithms

Collaborators / Publications

1. Dimitra Giannakopoulou, David H. Bushnell, Johann Schumann, Heinz Erzberger, Karen Heere: Formal testing for separation assurance. Ann. Math. Artif. Intell. 63(1): 5-30 (2011)

2. Dimitra Giannakopoulou, Falk Howar, Malte Isberner, Todd Lauderdale, Zvonimir Rakamaric, Vishwanath Raman: Taming test inputs for separation assurance. ASE 2014.

3. Marko Dimjasevic, Dimitra Giannakopoulou: Test-Case Generation for Runtime Analysis and Vice Versa: Verification of Aircraft Separation Assurance. ISSTA2015.

4. Christian von Essen, Dimitra Giannakopoulou: Probabilistic verification and synthesis of the next generation airborne collision avoidance system. STTT 18(2): 227-243 (2016). Extended version of TACAS 2014 paper awarded ETAPS 2014 EASST best paper.

5. Dimitra Giannakopoulou, Dennis Guck, Johann Schumann: Exploring Model Quality for ACAS X. FM 2016.